Boosting and Differential Privacy

Cynthia Dwork*, Guy N. Rothblum†, Salil Vadhan‡
*Microsoft Research, 1065 La Avenida, Mountain View, Ca 94043. dwork@microsoft.com
†Center for Computational Intractability and Department of Computer Science,
Princeton University, 35 Olden Street, Princeton, NJ 08544. rothblum@alum.mit.edu.
‡School of Engineering and Applied Sciences and Center for Research on Computation and Society,
Harvard University, 33 Oxford Street, Cambridge MA 02138. salil@seas.harvard.edu.

***Abstract*—Boosting is a general method for improving the accuracy of learning algorithms. We use boosting to construct improved *privacy-preserving synopses* of an input database. These are data structures that yield, for a given set $\mathcal{Q}$ of queries over an input database, reasonably accurate estimates of the responses to every query in $\mathcal{Q}$, even when the number of queries is much larger than the number of rows in the database. Given a *base synopsis generator* that takes a distribution on $\mathcal{Q}$ and produces a "weak" synopsis that yields "good" answers for a majority of the weight in $\mathcal{Q}$, our *Boosting for Queries* algorithm obtains a synopsis that is good for all of $\mathcal{Q}$. We ensure privacy for the rows of the database, but the boosting is performed on the *queries*. We also provide the first synopsis generators for arbitrary sets of arbitrary low-sensitivity queries, *i.e.*, queries whose answers do not vary much under the addition or deletion of a single row.**

In the execution of our algorithm certain tasks, each incurring some privacy loss, are performed many times. To analyze the cumulative privacy loss, we obtain an $O(\varepsilon^2)$ bound on the *expected* privacy loss from a single ε-differentially private mechanism. Combining this with evolution of confidence arguments from the literature, we get stronger bounds on the expected cumulative privacy loss due to multiple mechanisms, each of which provides ε-differential privacy or one of its relaxations, and each of which operates on (potentially) different, adaptively chosen, databases.

I. Background and Summary of Results

Boosting. Boosting is a general and widely used method for improving the accuracy of learning algorithms. (See [23] for an excellent survey.) Given a training set of labeled examples, $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where each x_i is drawn from an underlying distribution D on a universe X, and $y_i \in \{+1, -1\}$, a learning algorithm produces a hypothesis $h : X \to \{+1, -1\}$. Ideally, h will "describe" not just the given samples, but also the underlying distribution. The goal of boosting is to convert a weak learner, which produces a hypothesis that does just a little better than random guessing, into a strong, or very accurate, learner. Many boosting algorithms share the following basic structure. First, an initial (typically uniform) probability distribution is imposed on the sample set. Computation then proceeds in rounds. In each round t: (1) the base learner is run on the current distribution D_t, producing a classification hypothesis h_t; and (2) the hypotheses $h_1, \ldots, h_t$ are used to reweight the samples, defining D_{t+1}. The process halts either after a predetermined number of rounds or when an appropriate combining of the hypotheses is determined to be sufficiently accurate. The main design decisions are how to modify the probability distribution from one round to the next, and how to combine the hypotheses $\{h_t\}_{t=1,\ldots,T}$ to form a final output hypothesis.

Differential Privacy. Differential privacy is a notion of privacy tailored to private data analysis, where the goal is to learn information about the population as a whole, while protecting thea privacy of each individual. (See the surveys [7], [6].) Roughly speaking, differential privacy ensures that the system will behave in essentially the same fashion, independent of whether any individual opts in to, or out of, the database. Here, "behaves essentially the same way" means that the probability distribution over outputs of an analysis, where the probability space is the coin flips of the privacy mechanism, is essentially the same, independent of the presence or absence of any individual.

Early results on differential privacy showed how to accurately answer small to moderate numbers of *counting queries* of the form "How many rows in the database satisfy property P?" [12], [10]. Specifically, any set $\mathcal{Q}$ of counting queries could be answered in a differentially private manner with an accuracy of roughly $\sqrt{|\mathcal{Q}|}$, so for a database of size n, we can obtain nontrivial accuracy (namely, errors of magnitude $o(n)$) if $|\mathcal{Q}|$ is sufficiently smaller than n^2. (For simplicity, throughout the introduction, we hide dependence on parameters

Guy Rothblum's research is supported by NSF Grant CCF-0832797 and by a Computing Innovation Fellowship. Part of this work was done while he was visiting Microsoft Research.

Salil Vadhan's research is supported by NSF grant CNS-0831289.

other than $|\mathcal{Q}|$ and n.) In [10], [8], similar bounds were obtained for arbitrary *low-sensitivity* queries, that is queries whose output does not change much when one item is added or removed from the database.

A remarkable result of Blum, Ligett, and Roth [3] shows that differential privacy is possible even in cases when the number of *counting* queries is much larger than n^2. Specifically, given a set $\mathcal{Q}$ of counting queries, they show how to answer all the queries in $\mathcal{Q}$ within an error of roughly $n^{2/3} \cdot \log^{1/3} |\mathcal{Q}|$, which provides nontrivial accuracy provided that $|\mathcal{Q}|$ is sufficiently smaller than 2^n. In fact, they also provide a compact representation of all of these answers in the form of a *synthetic database*. This is a data structure that "looks like" a database, in that its rows are drawn from the same universe $\mathcal{X}$ from which the database rows are drawn. When appropriately scaled, the responses on the synthetic database to all the queries in $\mathcal{Q}$ approximate the answers to the same queries on the original database. Dwork *et al.* [11] improved the running time to $\mathrm{poly}(|\mathcal{X}|, |\mathcal{Q}|)$, where $\mathcal{X}$ is the universe of the database rows, and achieved an incomparable accuracy bound of roughly $\sqrt{n} \cdot |\mathcal{Q}|^{o(1)}$. Our Boosting for Queries algorithm, described next, is inspired by the differentially private synopsis generator of [11],

A compact representation for answers to a set $\mathcal{Q}$ of queries on a database x need not be in the form of a synthetic database; it can be an arbitrary data structure, which, when presented with any $q \in \mathcal{Q}$, returns an approximation to $q(x)$. We refer to such a data structure as a *synopsis* of the database x. General privacy-preserving synopses are of interest because they may be easier to construct than privacy-preserving synthetic databases. For example, there are stronger hardness results for constructing synthetic databases than are known for general privacy-preserving synopses [11], [26].

Summary of Results. Our principal result is a technique for generating privacy-preserving synopses for *any* set of low-sensitivity queries (not just counting queries). This is achieved by a novel use of boosting, together with the construction of an appropriate base synopsis generator.

Boosting for Queries. We introduce the notion of *boosting for queries*, where the items on which the boosting algorithm operates are the database queries, *i.e.,* the functions or analyses that the analyst wishes to evaluate on the database. We present an algorithm that, given a base synopsis generator that takes a distribution on $\mathcal{Q}$ and produces a "weak" synopsis that yields "good" answers for a majority of the weight in $\mathcal{Q}$, "boosts" it to obtain a synopsis that is good for all of $\mathcal{Q}$. Although the boosting is performed over the queries, the privacy is still for the rows of the database. The privacy challenge in boosting for queries come from the fact that each row in the database affects the answers to all the queries, and thus can potentially have a large influence on how the distribution on queries changes from one iteration to the next. Our algorithm is based on a variant of Freund and Schapire's Adaboost algorithm [13], due to Schapire and Singer [24]. To achieve privacy, we do not use a sharp threshold between "accurate" answers and "inaccurate" answers to decide whether to increase or decrease the weight on a query, but rather gradually change the weight as a function of how accurate the answer is. This way, no single row in the database has too much influence on the query distributions constructed by the boosting algorithm. The running time of the boosting procedure depends quasi-linearly on the number $|\mathcal{Q}|$ of queries and on the running time of the base synopsis generator. (In particular, it is independent of the data universe size $|\mathcal{X}|$.)

Base Synopsis Generators for Arbitrary Low-Sensitivity Queries and for Counting Queries. We provide a base synopsis generator for sets of arbitrary low-sensitivity queries. Applying boosting to it, we get the first privacy-preserving synopsis construction for arbitrary low-sensitivity queries. The accuracy of our boosted mechanism is roughly $\sqrt{n} \cdot \log^{3/2} |\mathcal{Q}|$. The running time of our base synopsis generator (and hence its boosted version) is large, namely $\mathrm{poly}(|\mathcal{Q}|, |\mathcal{X}|^n)$, where n is the size of the database.

For the special case of counting queries we can use a base synopsis generator from [11], and obtain accuracy roughly $\sqrt{n} \cdot \log |\mathcal{Q}|$ (improving on the bound of $\sqrt{n} \cdot |\mathcal{Q}|^{o(1)|}$ from [11]) with a running time of $\mathrm{poly}(|\mathcal{Q}|, |\mathcal{X}|)$. In subsequent work, Hardt and Rothblum [16] have obtained similar accuracy and running time (in fact with a better dependence on $|\mathcal{X}|$ than we have) for counting queries with an *interactive* mechanism that does not need to know all the queries in advance. (Previously, Roth and Roughgarden [22] gave such an interactive mechanism that achieved similar accuracy and efficiency to [3].)

Bounding Expected Privacy Loss and Composition Theorems. In the execution of our Boosting for Queries algorithm, certain tasks, each incurring some privacy loss, are performed many times. To analyze the cumulative privacy loss, we obtain an $O(\varepsilon^2)$ bound on the *expected* privacy loss from a single ε-differentially private mechanism. Combining this with evolution of confidence arguments from the literature [5], [12], we get stronger bounds on the expected cumulative privacy

loss due to multiple (say k) mechanisms, each providing ε-differential privacy or one of its relaxations (see Section II), and each operating on (potentially) different, adaptively chosen, databases. Roughly speaking, privacy will deteriorate as $\sqrt{k}\varepsilon + k\varepsilon^2$, rather than the worst-case $k\varepsilon$ known in the literature [10].

Boosting for People. As in previous works on differentially private learning [2], [17], we can also view the input database as a training set in a learning algorithm, where each row corresponds to an element in the training set. It is natural to try to combine learning and differential privacy: use learning theory to know what to compute on a database to understand the underlying population, and use the techniques for differential privacy to do this in a privacy-protective fashion, with small distortion when possible. In the full paper we present a differentially private boosting technique, in which privacy comes at little additional cost in accuracy. We call this *Boosting for People*, since rows corresponding to the data of individual people are the elements of interest. Further treatment of Boosting for People is omitted from these proceedings for lack of space.

II. Preliminaries and Definitions

We write $[n]$ for the set $\{1, 2, \ldots n\}$. Throughout the paper, we work with discrete probability spaces. Sometimes we will describe our algorithms as sampling from continuous distributions, but these should always be discretized to finite precision in some standard way (which we do not specify for sake of readability). For a discrete distribution (or random variable) X taking values in a set S, we denote by $x \leftarrow X$ the experiment of selecting $x \in S$ according to the distribution X. The *support* of X is denoted $\mathrm{Supp}(X) = \{x : \Pr[X = x] > 0\}$. A function $f : \mathcal{N} \to \mathbb{R}^+$ is *negligible* if for every constant c, $f(\kappa) < 1/\kappa^c$ for sufficiently large κ (i.e. $f(\kappa) = \kappa^{-\omega(1)}$). We write $\nu(\kappa)$ for an unspecified function that is negligible in κ.

In this work we deal with (non-interactive) methods for releasing information about a database. For a given database x, a (randomized) non-interactive database access mechanism $\mathcal{M}$ computes an output $\mathcal{M}(x)$ that can later be used to reconstruct information about x. We will be concerned with mechanisms $\mathcal{M}$ that are *private* according to various privacy notions described below.

We think of a database x as an ordered multiset of *rows*, each from a data universe X. Intuitively, each row contains the data of a single individual. We will often view a database of size n as a tuple $x \in X^n$ for some $n \in \mathcal{N}$ (the number of individuals whose data is in the database). We treat n as public information throughout.

We say databases x, x' of size n are *adjacent* if they agree on at least $n - 1$ rows. That is, two databases are adjacent if they are of the same size and their edit distance is at most 1. To handle worst case pairs of databases, our probabilities will be over the random choices made by the privacy mechanism.

Definition II.1 (Differential Privacy [10])**.** A randomized algorithm $\mathcal{M}$ is *ε-differentially private* if for all pairs of adjacent databases x, x', and for all sets $S \subseteq \mathrm{Supp}(\mathcal{M}(x)) \cup \mathrm{Supp}(\mathcal{M}(x'))$

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \cdot Pr[\mathcal{M}(x') \in S],$$

where the probabilities are over the coin flips of the algorithm $\mathcal{M}$.

Intuitively, this captures the idea that no individual's data has a large effect on the output distribution of the mechanism. Typically, we think of ε as a small constant. A basic example of a differentially private algorithm is the *Laplace mechanism* [10], which yields differentially private approximations to real-valued functions. Specifically, for a real-valued function f, the *(global) sensitivity of f* is the maximal absolute difference in its values on adjacent databases: $\max_{\text{adjacent } x,x'} |f(x) - f(x')|$. The *Laplace distribution* $\mathrm{Lap}(t)$ has density function $h(y) \propto e^{-y/t}$, has mean 0 and standard deviation t. We usually refer to the Laplace distribution over integers. Dwork *et al.* [10] showed that if a real-valued function f has global sensitivity at most s then the function $f(x) + \mathrm{Lap}(s/\varepsilon)$ is ε-differentially private. See [7], [6] for more properties and results concerning differential privacy.

We will also consider a relaxation of differential privacy, which allows us to ignore events of very low probability:

Definition II.2 ((ε, δ)-Differential Privacy [8])**.** A randomized algorithm $\mathcal{M}$ gives *(ε, δ)-differential privacy* if for all pairs of adjacent databases x and x' and all $S \subseteq \mathrm{Range}(\mathcal{M})$, $\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta$, where the probabilities are over the coin flips of the algorithm $\mathcal{M}$.

Observe that ε-differential privacy implies (ε, δ)-differential privacy. There is a simple example showing the converse implication does not hold. We note that there is another notion, known as (ε, δ)-probabilistic differential privacy [18], [14], which lies strictly between ε-differential privacy and (ε, δ)-differential privacy.

Auxiliary parameters and synopsis generators. Often our privacy mechanism $\mathcal{M}$ will take some auxiliary parameters w as input, in addition to the database x. For

example, w may specify a query q_w on the database x, or a collection $\mathcal{Q}_w$ of queries. The Mechanism $\mathcal{M}(w, x)$ might (respectively) respond with a differentially private approximation to $q_w(x)$ or to some or all of the queries in $\mathcal{Q}_w$. We say that a mechanism $\mathcal{M}(\cdot, \cdot)$ satisfies ε-differential privacy if for every w, $\mathcal{M}(w, \cdot)$ satisfies ε-differential privacy; and analogously for the other notions of privacy.

Another example of a parameter that may be included in w is a *security parameter* κ to govern how small $\delta = \delta(\kappa)$ should be. That is, $\mathcal{M}(\kappa, \cdot)$ should be $(\varepsilon, \delta(\kappa))$ differentially private for all κ. Typically, and throughout this paper, we require that δ be a negligible function in κ, i.e. $\delta = \kappa^{-\omega(1)}$. Thus, we think of δ as being cryptographically small, whereas ε is typcially thought of as a moderately small constant.

In the case where the auxiliary parameter w specifies a collection $\mathcal{Q}_w = \{q : X^n \to \mathbb{R}\}$ of queries, we call the mechanism $\mathcal{M}$ a *synopsis generator*. A synopsis generator outputs a (differentially private) synopsis $\mathcal{A}$ which can be used to compute answers to all the queries in $\mathcal{Q}_w$. I.e., we require that there exists a reconstruction procedure R such that for each input v specifying a query $q_v \in \mathcal{Q}_w$, the reconstruction procedure outputs $R(\mathcal{A}, v) \in \mathbb{R}$. Typically, we will reuqire that with high probability $\mathcal{M}$ produces a synopsis A s.t. the reconstruction procedure, using $\mathcal{A}$, computes accurate answers. I.e., for all or most (weighted by some distribution) of the queries $q_v \in \mathcal{Q}_w$, the error $|R(\mathcal{A}, v) - q_v(x)|$ will be bounded. We will occasionally abuse notation and refer to the reconstruction procedure taking as input the actual query q (rather than some representation v of it), and outputting $R(\mathcal{A}, q)$.

Divergence. One can rephrase the privacy notions above in terms of distance measures between distributions. In the fractional quantities below, if the denominator is 0, then we define the value of the fraction to be infinite (the numerators will always be positive). For two discrete random variables Y and Z, their *KL divergence* (i.e. *relative entropy*) is defined to be

$$D(Y||Z) \stackrel{\text{def}}{=} \mathrm{E}_{y \leftarrow Y}\left[\ln \frac{\Pr[Y = y]}{\Pr[Z = y]}\right].$$

It is known that $D(Y||Z) \geq 0$, with equality iff Y and Z are identically distributed. (However, D is not symmetric, does not satisfy the triangle inequality, and can even be infinite, specifically when $\mathrm{Supp}(Y)$ is not contained in $\mathrm{Supp}(Z)$.) We can obtain a worst-case analogue of KL divergence by taking a maximum instead of an expectation (analogous to how min-entropy relates to Shannon entropy):

$$\begin{aligned} D_\infty(Y||Z) &\stackrel{\text{def}}{=} \max_{y \in \mathrm{Supp}(Y)}\left[\ln \frac{\Pr[Y = y]}{\Pr[Z = y]}\right] \\ &= \max_{S \subseteq \mathrm{Supp}(Y)}\left[\ln \frac{\Pr[Y \in S]}{\Pr[Z \in S]}\right]. \end{aligned}$$

We refer to $D_\infty(Y||Z)$ as the *max-divergence* of Y and Z. This is a rather brittle measure, in that a change in even a small portion of probability space can affect $D_\infty(Y||Z)$ dramatically. Thus it is natural to allow ourselves to discard a small fraction of the probability space, leading to *δ-approximate max-divergence*, which we define by:

$$D_\infty^\delta(Y||Z) \stackrel{\text{def}}{=} \max_S\left[\ln \frac{\Pr[Y \in S] - \delta}{\Pr[Z \in S]}\right],$$

where $S \subseteq \mathrm{Supp}(Y) : \Pr[Y \in S] > \delta$.

We have: (i) A randomized algorithm $\mathcal{M}$ gives ε-differential privacy iff for all pairs of adjacent databases x and x', we have $D_\infty(\mathcal{M}(x)||\mathcal{M}(x')) \leq \varepsilon$. And (ii) A randomized algorithm $\mathcal{M}$ gives (ε, δ)-differential privacy iff for all pairs of adjacent databases x and x', we have $D_\infty^\delta(\mathcal{M}(x)||\mathcal{M}(x')) \leq \varepsilon$.

One other distance measure that will be useful is *statistical distance* between two random variables Y and Z, defined as

$$\Delta(Y, Z) \stackrel{\text{def}}{=} \max_S |\Pr[Y \in S] - \Pr[Z \in S]|.$$

We say that Y and Z are *δ-close* if $\Delta(Y, Z) \leq \delta$.

III. Composition Theorems

In this section, we provide general results about the composition of differentially private mechanisms. There are several reasons for studying composition (analogous to the reasons that researchers have studied the composition of cryptographic protocols):

1) Composition can be used for the modular design of complex private mechanisms from simpler ones.
2) Composition models repeated use of the *same* mechanism on the *same database*; we want to be assured that its privacy guarantees will not degrade too much.
3) Composition models the interaction between many *different* privacy mechanisms. If Alice's data is used in many differentially private data releases over her lifetime, involving different databases and different mechanisms, we still would like to assure her that her privacy will not be compromised too much.

Previous composition results for differential privacy have primarily been concerned with the first two items. Here we introduce a form of composition that captures the very general setting suggested in Item 3, and prove composition results for it. Moreover, we revisit the "evolution of confidence" arguments due to Dinur, Dwork, and Nissim [5], [12] and show that, for achieving (ε, δ) differential privacy of k-fold composition, the privacy parameter ε degrades as $k\varepsilon^2 + \sqrt{k}\varepsilon$ rather than linearly as $k\varepsilon$. Previously this was shown only for specific mechanisms, and only when all k applications were on the same database.

A. Modeling Composition

We want to model composition where the adversary can adaptively affect the databases being input to future mechanisms, as well as the queries to those mechanisms. We do this by introducing a differentially private analogue of the "left or right" notion of security for encryption schemes, due to Bellare, Desai, Jokipii, and Rogaway [1]. Let $\mathbb{M}$ be a family of database access mechanisms. (For example $\mathbb{M}$ could be the set of all ε-differentially private mechanisms.) For a probabilistic adversary A, we consider two experiments, Experiment 0 and Experiment 1, defined as follows.

k-fold Composition Experiment b for mechanism family $\mathbb{M}$ and adversary A:[1] For $i = 1, \ldots, k$:

1) A outputs two adjacent databases x_i^0 and x_i^1, a mechanism $\mathcal{M}_i \in \mathbb{M}$, and parameters w_i.
2) A receives $y_i \leftarrow \mathcal{M}_i(w_i, x_{i,b})$.

We allow the adversary A above to be stateful throughout the experiment, and thus it may choose the databases, mechanisms, and the parameters adaptively depending on the outputs of previous mechanisms. We define A's *view* of the experiment to be A's coin tosses and all of the mechanism outputs $(y_1, \ldots, y_k)$. (The x_i^j's, $\mathcal{M}_i$'s, and w_i's can all be reconstructed from these.)

For intuition, consider an adversary who always chooses x_i^0 to hold Bob's data and x_i^1 to differ only in that Bob's data is replaced with junk. Then experiment 0 can be thought of as the "real world," where Bob allows his data to be used in many data releases, and Experiment 1 as an "ideal world," where the outcomes of these data releases do not depend on Bob's data. Our definitions of privacy still require these two experiments to be "close" to each other, in the same way as required by the definitions of differential privacy. The intuitive guarantee to Bob is that the adversary "can't tell", given the output of all k mechanisms, whether Bob's data was ever used.

[1] We remark that allowing both a mechanism family $\mathbb{M}$ and auxiliary parameters w is redundant. The parameters w can be removed by expanding the family $\mathbb{M}$ to $\mathbb{M}' = \{\mathcal{M}(\cdot, w)\}_{\mathcal{M} \in \mathbb{M}, w}$, and conversely, we can use the parameters to collapse $\mathbb{M}$ to a single mechanism $\mathcal{M}^*(x, (\mathcal{M}, w))$ that outputs $\mathcal{M}(x, w)$ if $\mathcal{M} \in \mathbb{M}$ and outputs $\perp$ otherwise.

Definition III.1. We say that the family $\mathbb{M}$ of database access mechanisms satisfies *ε-differential privacy under k-fold adaptive composition* if for every adversary A, we have $D_\infty(V^0 || V^1) \leq \varepsilon$ where V^b denotes the view of A in k-fold Composition Experiment b above.

(ε, δ)-differential privacy under k-fold adaptive composition instead requires that $D_\infty^\delta(V^0 || V^1) \leq \varepsilon$.

B. Composition Theorems

Speaking colloquially, it is already known that when we compose differentially private mechanisms "the epsilons and deltas add up" (cf. [10] for ε-differential privacy and [9], [19], [8] for (ε, δ)-differential privacy). This also extends to our general model of composition:

Theorem III.1. *For every $\varepsilon, \delta \geq 0$ and $k \in \mathbb{N}$,*

1) *The family of ε-differentially private mechanisms satisfies $k\varepsilon$-differential privacy under k-fold adaptive composition.*
2) *The family of (ε, δ)-differentially private mechanisms satisfies $(k\varepsilon, k\delta)$-differential privacy under k-fold adaptive composition.*

Thus, if Bob's data is to be involved in k data releases over his lifetime, the above theorem suggests that he should require both ε and δ to be smaller than $1/k$. For δ, this is not problematic as we anyhow take δ to be very small (negligible). But requiring ε to be this small can cause a significant price in utility (e.g. expected error $\Theta(k)$ is incurred in the Laplace mechanism).

For specific mechanisms applied on a single database, there are "evolution of confidence" arguments due to Dinur, Dwork, and Nissim [5], [12] showing that the privacy parameter need only deteriorate like $\sqrt{k}$ if we are willing to tolerate a (negligible) loss in δ (for $k < 1/\varepsilon^2$). Here we generalize those arguments to arbitrary differentially private mechanisms, as well as to the general form of composition described above.

Our key technical contribution shows that a bound of ε on the *worst-case* privacy loss (as captured by max-divergence) implies a bound of $O(\varepsilon^2)$ on the *expected* privacy loss (as captured by KL divergence).

Lemma III.2. *Suppose that random variables Y and Z satisfy $D_\infty(Y||Z) \leq \varepsilon$ and $D_\infty(Z||Y) \leq \varepsilon$. Then $D(Y||Z) \leq \varepsilon \cdot (e^\varepsilon - 1)$.*

Note that $e^{\varepsilon} \leq 1 + 2\varepsilon$ for $\varepsilon \in [0, 1]$, so we get a bound of $D(Y||Z) \leq 2\varepsilon^2$. Next we apply concentration bounds to show that with high probability the "privacy loss" is close to the expectation (which, by the above, is $O(\varepsilon^2 k)$ rather than εk), and thus giving us (ε', δ) for privacy for $\varepsilon' \ll k\varepsilon$ for composing k ε-differentially private mechanisms. Due to the adaptivity of the adversary, the outputs of mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are not independent. Following [5], [12], we use Azuma's Inequality to establish the concentration we want.

For composition of (ε, δ)-differential privacy, we use a characterization of approximate max-divergence in terms of statistical difference and standard max-divergence to reduce the analysis to the same situation as in the case of ε-differential privacy. Specifically, the characterization says that if $D_{\infty}^{\delta}(Y||Z) \leq \varepsilon$ and $D_{\infty}^{\delta}(Z||Y) \leq \varepsilon$, then Y is δ-close to a random variable Y' such that $D_{\infty}(Y'||Z) \leq \varepsilon$ and $D_{\infty}(Z||Y') \leq \varepsilon$. (This can be viewed as an information-theoretic analogue of the "dense model theorems" of [15], [25], [21], [20].)

This yields the following theorem (which includes the case of ε-differential privacy by setting $\delta = 0$):

Theorem III.3. *For every $\varepsilon > 0, \delta, \delta' > 0$, and $k \in \mathbb{N}$, the class of (ε, δ)-differentially private mechanisms is $(\varepsilon', k\delta + \delta')$-differentially private under k-fold adaptive composition, for*

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k \cdot \varepsilon\varepsilon_0,$$

where $\varepsilon_0 = e^{\varepsilon} - 1$.

IV. Boosting for Queries

In this section we present a query-boosting algorithm for arbitrary low-sensitivity queries. The algorithm takes a weak, sometimes-accurate, differentially private *"base synopsis generator"* (i.e. one that outputs privacy-preserving answers to most of the mass of a given query distribution), and "boosts" it to get an always (or often) accurate synopsis generator. This is done while maintaining differential privacy. We begin with a more formal treatment of the setup, follow with an overview, and then present the query-boosting algorithm in Figure 1. Its accuracy and privacy guarantees are described in Theorem IV.1.

The Setup. Recall the framework outlined in the introduction: We are given a database $x \in X^n$ and a class $\mathcal{Q} = \{q : X^* \to \mathbb{R}\}$ of queries, and we wish to answer all the queries in a differentially private manner.

Our databases will be of size n, with rows, or data elements, drawn from a data universe X. We are given a query set $\mathcal{Q} = \{q : X^n \to \mathbb{R}\}$ of real-valued queries.

Definition IV.1 (sensitivity of a query family)**.** The sensitivity of the query family $\mathcal{Q}$ will be denoted by $\rho = \rho(n)$. It is defined to be $\rho(n) = \max\{|q(x) - q(x')| : q \in Q,$ adjacent $x, x' \in X^n\}$ (the maximum over all queries in the family of their sensitivities).

High-sensitivity queries are inherently problematic for preserving privacy, as, by definition, changing the value of one data element can radically affect the outcome of the query. Responses to such queries therefore seem to require large distortion in order to ensure privacy. We are therefore interested in small values of ρ.

Definition IV.2 ($(k, \lambda, \eta, \beta)$-base synopsis generator)**.** For a fixed database size n, data universe X and query set $\mathcal{Q}$, consider a synopsis generator $\mathcal{M}$, that takes as input k queries from the query family $\mathcal{Q}$ and outputs a synopsis. We say that $\mathcal{M}$ is a $(k, \lambda, \eta, \beta)$-*base synopsis generator* if for any distribution D on $\mathcal{Q}$, when $\mathcal{M}$ is activated on a database $x \in X^n$ and on k queries sampled independently from D, with all but β probability (over the k queries drawn from D and the coins of $\mathcal{M}$) the synopsis that $\mathcal{M}$ outputs is λ-accurate (w.r.t x) for a $(1/2 + \eta)$-fraction of mass of $\mathcal{Q}$ as weighted by D.

We will be interested in differentially private base synopsis generators. The goal is to *"boost"* such a base synopsis generator into a strong synopsis generator that, with high probability, accurately answers all or almost all of the queries (on the same database x), while still preserving (ε, δ)-differential privacy. Given these parameters, we would like to optimize the accuracy of the resulting synopsis generator (aiming to stay as close as possible to λ-accuracy). We let μ denote the additional error incurred by the resulting synopsis generator.

Overview. We use the base synopsis generator, running it repeatedly for T rounds, generating a synopsis $\mathcal{A}_t$ in round t. We maintain a distribution $\mathcal{D}$ on the queries (initially this distribution is uniform), and reweight so that in each round queries for which the base synopsis generator failed to produce an accurate answer get higher probability. The objects $\mathcal{A}$ are combined by taking the median: given $\mathcal{A}_1, \ldots, \mathcal{A}_T$, the quantity $q(DB)$ is estimated by taking the approximate values for $q(DB)$ computed using each of the $\mathcal{A}_i$, and then computing their median. The algorithm will run for a fixed number T of rounds (roughly $T \in O(\log(|Q|))$.

There are "standard" methods for updating $\mathcal{D}$, for example, increasing the weight of poorly handled elements, in our case, queries, by a factor of e and

decreasing the weight of well-handled elements by the same factor. However, we need to protect the privacy of the database rows, and a single database row can have a substantial effect on $\mathcal{D}$ if it makes the difference between being "well approximated" or "poorly approximated" for many queries. We therefore need to mitigate the effect of any database row. This is done by attenuating the re-weighting procedure. Instead of always using a fixed ratio either for increasing the weight (when the answer is "accurate") or decreasing it (when it is not), we set separate thresholds for "accuracy" and "inaccuracy". Queries for which the error is below or above these thresholds have their weight decreased or increased (respectively) by a fixed factor. For queries whose error lies between these two thresholds, we scale the (logarithm of the) weight change linearly. The attenuated scaling reduces the effect of any individual on a the re-weighting of any query. This is because any individual can only affect the true answer to a given query, and thus also the accuracy of the base synopsis generator's output, by a small amount.

The larger the gap between the "accurate" and "inaccurate" thresholds, the smaller the effect of each individual on a query's weight can be. This means that larger gaps are better for privacy. For accuracy, however, large gaps are bad. If the inaccuracy threshold is large, we can only guarantee that queries for which the base synopsis generator is very inaccurate have their weight increased during re-weighting. This degrades the accuracy guarantee of the boosted synopsis generator: it is roughly equal to the "inaccuracy" threshold.

The query-boosting algorithm is general. It can be used for any class of queries (not only counting queries) and any differentially private base synopsis generator. The running time is, to a large extent, inherited from the base synopsis generator, which need only be run roughly $\log|\mathcal{Q}|$ times (assuming it has constant advantage η over $1/2$). The booster invests additional time that is quasi-linear in $|\mathcal{Q}|$ in the boosting process, and in particular its running time does not depend directly on the size of the data universe from which data items come.

Notation. Throughout the algorithm's operation, we keep track of several variables (explicitly or implicitly). Variables indexed by $q \in \mathcal{Q}$ hold information pertaining to query q in the query set. We run T rounds of boosting. Variables indexed by $t \in [T]$, usually computed in round t, will be used to construct the distribution D_{t+1} used for sampling in time period $t+1$. For a predicate P we use $[[P]]$ to denote 1 if the predicate is true and 0 if it is false.

Theorem IV.1 (Boosting for Queries)**.** *Let $\mathcal{Q}$ be a query family with sensitivity at most ρ. For an appropriate setting of parameters, and with $T = O(\log|Q|/\eta^2)$ rounds, the algorithm of Figure 1 is an accurate and differentially private query-boosting algorithm:*

1) *When instantiated with a $(k, \lambda, \eta, \exp(-\kappa))$-base synopsis generator, the output of the boosting algorithm gives $(\lambda + \mu)$-accurate answers to* all *the queries in $\mathcal{Q}$ with probability at least $1 - T \cdot \exp(-\kappa)$.*
2) *For $\varepsilon \in [0, 1]$, if the base synopsis generator is $(\varepsilon_{base}, \delta_{base})$-differentially private and $\mu = O((\log^{3/2}|Q| \cdot \sqrt{k} \cdot \sqrt{\kappa} \cdot \rho)/(\varepsilon \cdot \eta^2))$, then the boosting algorithm is $((\varepsilon + T \cdot \varepsilon_{base}), T \cdot (k \cdot \exp(-\kappa) + \delta_{base}))$-differentially private.*

The proof of accuracy follows the structure of Adaboost's accuracy analysis in [24]. The proof of privacy considers adjacent databases x and x'. In each of the T rounds of boosting, fixing the past answers $\mathcal{A}_1, \ldots, \mathcal{A}_t$, we use D_{t+1} to denote the next round's distribution computed using database x and D'_{t+1} to denote the next round's distribution computed using database x'. For the various quantities computed by the algorithm in round t using database x, such as $u_{t,q}$ or Z_t, we use $u_{t,q}{}'$ or Z'_t to denote their counterparts computed using database x'.

In Claim IV.2 below, we show that (fixing the previously computed $\mathcal{A}_j$'s), for adjacent x, x', the max-divergence of the distributions D_{t+} and D'_{t+1} is bounded.

Claim IV.2. *Let x and x' be adjacent databases. After $t \in [T]$ rounds of boosting, fix $\mathcal{A}_1, \ldots, \mathcal{A}_t$. Let D_{t+1} and D'_{t+1} be the corresponding distributions for the $(t+1)$-th round of boosting. Then $D_\infty(D_{t+1}||D'_{t+1}) \leq 4\alpha \cdot T \cdot \rho/\mu$.*

To analyze the effect of T iterations we use the composition model developed in Section III-A. A natural approach would be to analyze the privacy loss incurred by a single iteration of the algorithm, and then use the composition theorems to bound the total privacy loss. However, the algorithm maintains state between iterations in the form of the distributions D_t, $t = 1, 2, \ldots, T$. This can be avoided by having the algorithm release the hypotheses A_t as, given the hypotheses $\mathcal{A}_1, \mathcal{A}_2, \ldots, A_{t-1}$ (which are in any case made public at the end of the algorithm), a straightforward and deterministic computation on the database yields D_t.

In fact, a different decomposition of the steps in the algorithm's T rounds yields a better privacy bound. Specifically, we we separately analyze the privacy losses due to the T executions of the base generator and the

Boosting for Queries$(k, \lambda, \eta, \rho, \mu, T)$
Given: database $x \in X^n$, query set $\mathcal{Q}$, where each $q \in \mathcal{Q}$ is a function $q : X^n \to \mathbb{R}$ with sensitivity at most ρ.
Initialize D_1 to be the uniform distribution over $\mathcal{Q}$.
For $t = 1, \ldots, T$:

1) Sample a sequence $S_t \subseteq \mathcal{Q}$ of k samples chosen independently and at random from D_t.
   Run the base synopsis generator to compute an answer data structure $\mathcal{A}_t : \mathcal{Q} \to \mathbb{R}$ that is w.h.p. accurate for at least $(1 + \eta)/2$ of the mass of D_t.
2) Reweight the queries. For each $q \in \mathcal{Q}$:
   a) If $\mathcal{A}_t$ is λ-accurate, then $a_{t,q} \leftarrow 1$
      If $\mathcal{A}_t$ is $(\lambda + \mu)$-inaccurate, then $a_{t,q} \leftarrow -1$
      Otherwise, let $d_{q,t} = |q(x) - \mathcal{A}_t(q)|$ be the error of $\mathcal{A}_t$ (between λ and $\lambda + \mu$) on q:
      $$a_{t,q} \leftarrow 1 - 2(d_{t,q} - \lambda)/\mu$$
   b) $u_{t,q} \leftarrow \exp(-\alpha \cdot \sum_{j=1}^{t} a_{j,q})$, where $\alpha = 1/2 \ln((1 + 2\eta)/(1 - 2\eta))$.
3) Renormalize:
   $$Z_t \leftarrow \sum_{q \in \mathcal{Q}} u_{t,q}$$
   $$D_{t+1}[q] = u_{t,q}/Z_t$$

Output the data structure $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_T)$. For $q \in \mathcal{Q}$:
$$\mathcal{A}(q) = median\{\mathcal{A}_1(q), \ldots, \mathcal{A}_T(q)\}$$

Figure 1. **Boosting for Queries** (a variant of AdaBoost [24])

Tk samples taken from the distributions $D_1, \ldots, D_T$, and add the results together via Theorem III.1.

In more detail, each database access takes as input the synopses generated in previous rounds: there are T executions of the base synopsis generator, a mechanism which is $(\varepsilon_{base}, \delta_{base})$-differentially private, and we also sample $k \cdot T$ times from the distributions $\{D_t\}_{t=1}^{T}$, each such sample is a $(4\alpha \cdot T \cdot \rho/\mu)$-differentially private mechanism (by Claim IV.2). Using the composition theorems (Theorems III.1 and III.3), we conclude that the boosting algorithm in its entirety is: $(\varepsilon_{boost}, \delta_{boost})$-differentially private, where

$$\varepsilon_{boost} = (T \cdot \varepsilon_{base}) + O(\sqrt{T \cdot k \cdot \kappa} \cdot ((\alpha \cdot T \cdot \rho)/\mu))$$
$$\delta_{boost} = T \cdot (\delta_{base} + k \cdot \exp(-\kappa)))$$

To get the parameters claimed in the theorem statement we can take:

$$\mu = O((T^{3/2} \cdot \sqrt{k} \cdot \sqrt{\kappa} \cdot \alpha \cdot \rho)/\varepsilon)$$

The algorithm sets $\alpha = (1/2) \ln((1 + 2\eta)/(1 - 2\eta)) = O(\eta)$. For accuracy, we need the number of rounds to be $T = O(\log |Q|/\eta^2)$. This yields the accuracy bound claimed in the theorem.

V. Applications of Boosting for Queries

In this section we detail applications of the query boosting algorithm. We construct base synopsis generators using a generalization argument due to [11], see Section V-A. In Section V-B we use this bound to construct base synopsis generators for arbitrary low-sensitivity queries (with high running time) and counting queries (with better running time and accuracy). Plugging these base synopsis generators into the boosting for queries algorithm yields new boosted mechanisms for answering large numbers of low-sensitivity queries, we detail the parameters that are obtained in Section V-C.

A. A Generalization Bound

We have a distribution D over a large set $\mathcal{Q}$ of queries to be approximated. If we wanted accurate and differentially private answers to *all* the queries in $\mathcal{Q}$, then the standard approach of adding (say Gaussian) noise would yield error roughly proportional to $\sqrt{|\mathcal{Q}|}$. If, however, we only want accurate answers to *most* of the queries in $\mathcal{Q}$ (weighted by the distribution D), we can (in some settings) employ a generalization argument of [11]. They show that if a small enough synopsis (a synthetic database or some other data structure) gives good enough approximations to the answers of a *randomly selected* subset $S \subset \mathcal{Q}$ of queries sampled by D, then with high probability (over the choice of S) it also gives good approximations to the answers to *most* queries in $\mathcal{Q}$ (weighted by D). As they showed, this observation can lead to significantly better approximations.

We use $R(y,q)$ to denote the answer given by the synopsis y (when used as input for the reconstruction procedure) on query q. Formally, we say that a synopsis y *λ-fits* a database x w.r.t a set S of queries if $\max_{q\in S}|R(y,q)-q(x)| \leq \lambda$. The generalization bound shows that if y λ-fits x w.r.t a large enough (larger than $|y|$) randomly chosen set S of queries sampled from a distribution D, then w.h.p y λ-fits x for *most* of D's query mass.

The proof of Lemma V.1 is a generalization to arbitrary distributions of the statement in [11] (that statement was made for the uniform distribution).

Lemma V.1. [11] *Let $\mathcal{D}$ be an arbitrary distribution on a query set $\mathcal{Q} = \{q : X^* \to \mathbb{R}\}$. For all $m \in \mathcal{N}$, $\beta \in (0,1)$, $\eta \in [0,1/2)$, take $a = 2(\log(1/\beta)+m)/(m\cdot(1-2\eta))$. Then with probability at least $1-\beta$ over the choice of $S \sim D^{a\cdot m}$, every synopsis y of size at most m bits that λ-fits x w.r.t. the query set S, also λ-fits x w.r.t. at least a $(1/2+\eta)$-fraction of D.*

B. Base Synopsis Generators

We now use the generalization bound to construct privacy-preserving base synopsis generators. Given a query distribution D, the idea is to sample a small set of queries from D, answer them by adding independent noise, and then output a synopsis (a synthetic database of bounded size) that "fits" the noisy answers. By Lemma V.1, if the number of queries we sampled was large enough (larger than the synopsis size), this synopsis will also (w.h.p.) "accurately answer" a random query sampled from D. As mentioned above, here our synopses will be synthetic databases. We show how to find a database that fits the noisy answers for arbitrary queries (in large time), and for counting queries (more efficiently, following [11]). These base synopses appear in the theorems below.

Theorem V.2 (Base Synopsis Generator for Arbitrary Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of queries of sensitivity at most ρ, for any $\varepsilon \in [0,1]$ and $\kappa > 0$, there exists a $(k, \lambda, \eta = 1/3, \beta = \exp(-\kappa))$-base synopsis generator for Q, where $k = O(n\cdot\log(|X|)\cdot\kappa)$ and $\lambda = \tilde{O}((\sqrt{n\cdot\log|X|}\cdot\rho\cdot\kappa^{3/2})/\varepsilon)$ Moreover, this base synopsis generator is $(\varepsilon, \exp(-\kappa))$-differentially private. Its running time is $|X|^n\cdot\mathrm{poly}(n,\kappa,\log(1/\varepsilon))$.*

Theorem V.3 (Base Synopsis Generator for Counting Queries [11])**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of counting queries (with sensitivity at most $1/n$), for any $\varepsilon \in [0,1]$ and $\kappa > 0$, there exists a $(k, \lambda, \eta = 1/3, \beta = \exp(-\kappa))$-base synopsis generator for Q, where $k = \tilde{O}(n\cdot\log(|X|)\cdot\kappa/\log|Q|)$ and $\lambda = \tilde{O}((\sqrt{\log|Q|}+\sqrt{\log|X|}\cdot\kappa^{3/2})/(\varepsilon\cdot\sqrt{n}))$. Moreover, this bases synopsis generator is $(\varepsilon,\exp(-\kappa))$-differentially private. Its running time is $\mathrm{poly}(|X|,n,\kappa,\log(1/\varepsilon))$.*

C. Putting It Together

Plugging the base synopsis generators of Section V-B together into the query boosting algorithm, we obtain the following boosted algorithms for answering large sets of low-sensitivity queries:

Theorem V.4 (Boosted Synopsis Generator for Arbitrary Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of queries of sensitivity at most ρ, for any $\varepsilon \in [0,1]$ and $\kappa > 0$, there exists an $(\varepsilon,\exp(-\kappa))$-differentially private synopsis generator for Q. With all but $\exp(-\kappa)$ probability its answers are λ-accurate for every query in $\mathcal{Q}$, where*

$$\lambda = \tilde{O}\left(\frac{\sqrt{n\cdot\log|X|}\cdot\rho\cdot\log^{3/2}|\mathcal{Q}|\cdot\kappa^{3/2}}{\varepsilon}\right)$$

Its running time is $|X|^n\cdot|Q|\cdot\mathrm{poly}(n,\kappa,\log(1/\varepsilon))$.

Theorem V.5 (Boosted Synopsis Generator for Counting Queries)**.** *For any data universe X, database size n, and class $\mathcal{Q} : \{X^n \to \mathbb{R}\}$ of counting queries (with sensitivity at most $1/n$), for any $\varepsilon \in [0,1]$ and $\kappa > 0$, there exists an $(\varepsilon,\exp(-\kappa))$-differentially private synopsis generator for Q. With all but $\exp(-\kappa)$ probability its answers are λ-accurate for every query in $\mathcal{Q}$, where*

$$\lambda = \tilde{O}\left(\frac{\sqrt{\log|X|}\cdot\log|\mathcal{Q}|\cdot\kappa^{3/2}}{\varepsilon\cdot\sqrt{n}}\right)$$

Its running time is $\mathrm{poly}(|X|,|Q|,n,\kappa,\log(1/\varepsilon))$.

REFERENCES

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, pages 26–45, 1998.

[2] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, June 2005.

[3] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th ACM SIGACT Symposium on Thoery of Computing*, 2008.

[4] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2006.

[5] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[6] C. Dwork. A firm foundation for private data analysis. *Communications of the ACM (to appear)*.

[7] C. Dwork. An ad omnia approach to defining and achieving private data analysis. In F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, editors, *Privacy, Security, and Trust in KDD, First ACM SIGKDD International (PinKDD), Revised Selected Papers*, volume 4890 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2007.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: privacy via distributed noise generation. In *Advances in Cryptology: Proceedings of EUROCRYPT*, pages 486–503, 2006.

[9] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the 2009 International ACM Symposium on Theory of Computing (STOC)*, 2009.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[11] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[12] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of CRYPTO 2004*, volume 3152, pages 528–544, 2004.

[13] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1, part 2):119–139, 1997.

[14] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Privacy in search logs. *CoRR*, abs/0904.0682, 2009.

[15] B. Green and T. Tao. The primes contain arbitrarily long arithmetic progressions. *Annals of Mathematics*, 167:481–547, 2008.

[16] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for interactive privacy-preserving data analysis. *FOCS (to appear)*, 2010.

[17] S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proceedings of FOCS 2008*, 2008.

[18] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vihuber. Privacy: From theory to practice on the map. In *Proc. ICDE*, 2008.

[19] I. Mironov. Personal communication, 2009.

[20] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In S. Halevi, editor, *Advances in Cryptology—CRYPTO '09*, volume 5677 of *Lecture Notes in Computer Science*, pages 126–142. Springer-Verlag, 16–20 August 2009.

[21] O. Reingold, L. Trevisan, M. Tulsiani, and S. Vadhan. Dense subsets of pseudorandom sets. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 76–85. IEEE, 26–28 October 2008.

[22] A. Roth and T. Roughgarden. The median mechanism: Interactive and efficient privacy with multiple queries. In *Proc. of STOC*, 2010.

[23] R. Schapire. The boosting approach to machine learning: An overview. In *D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, Nonlinear Estimation and Classification*. Springer, 2003.

[24] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 39:297–336, 1999.

[25] T. Tao and T. Ziegler. The primes contain arbitrarily long polynomial progressions. *Acta Mathematica*, 201:213–305, 2008.

[26] J. Ullman and S. Vadhan. PCPs and the hardness of generating synthetic data. Technical Report TR10-017, Electronic Colloquium on Computational Complexity, February 2010.